Kapitola: Nie-primitívna rekurzia

Z minula nám zostala nezodpovedaná otázka: existuje totálna funkcia, ktorú by sme považovali za algoritmicky vypočítateľnú, ale pritom by nebola primitívne rekurzívna?

A tiež je tu s tým súvisiaca podotázka: existuje program s while-cyklami a/lebo všeobecnou rekurziou, ktorý bude počítať totálnu funkciu ktorá nebude primitívne rekurzívna?

Čo už vieme povedať? Veta o primitívne rekurzívnej časovej zložitosti nám hovorí, že ak aj taká funkcia existuje, bude sa počítať veľmi, veľmi, veľmi zložito.

1.1 Ackermann

Najskôr sa k odpovedi na otázku zo začiatku kapitoly prepracujeme cestou, ktorou bola táto otázka zodpovedaná prvýkrát – v dobe, kedy o programoch v dnešnej podobe nebolo ani chýru, ani slychu.

Vieme už, že vieme spraviť hovadsky rýchlo rastúce primitívne rekurzívne funkcie: zo successora sme naskladali sčítanie, zo sčítania násobenie, z násobenia vieme naskladať mocninové veže, a tie už rastú hovadsky rýchlo. A komu nestačí, nech zoberie tie a spraví ešte jeden krok.

Dostávame tak obrovskú rýchlosť rastu, že je prirodzená otázka: existuje ľubovoľne rýchlo rastúca primitívne rekurzívna funkcia? Ak by existovala, bolo by to vynikajúce – z vety o primitívne rekurzívnej časovej zložitosti by zjavne plynulo, že každá totálna funkcia počítaná ľubovoľným algoritmom je primitívne rekurzívna.

Ale ako sa vraví v našich končinách: hovno, hovno, zlatá rybka.

Začneme príkladom jednej funkcie, ktorá:

- je veľmi jednoducho definovateľná
- na základe definície je algoritmicky vypočítateľná
- rastie rýchlejšie ako hociktorá primitívne rekurzívna funkcia

Ackermannova fcia 2 parametrov:

$$\begin{aligned} A(0,n) &= s(n) \\ A(m+1,0) &= A(m,1) \\ A(m+1,n+1) &= A(m,A(m+1,n)) \end{aligned}$$

Nejaké špeciálne prípady:

$$\begin{aligned} A(0,n) &= n+1 \\ A(1,n) &= n+2 \\ A(2,n) &= 2n+3 \\ A(3,n) &= 2^{n+3}-3 \\ A(4,n) &= \underbrace{2^{2^{\cdot^{\cdot^{2}}}}}_{n+3} - 3 \end{aligned}$$

Konkrétne teda

$$A(4,2) = 2^{2^{2^{2^2}}} - 3 = 2^{2^{16}} - 3 = 2^{65\,536} - 3 \sim 2 \cdot 10^{19\,728}$$

Inými slovami, $A(4,2)$ má asi dvadsaťtisíc cifier.

Základná myšlienka definície je, že táto funkcia v princípe zodpovedá našej postupnosti „čoraz viac rastúcich funkcií“ – $A(0,n)$ je analógiou successora, $A(1,n)$ sčítania, $A(2,n)$ násobenia, $A(3,n)$ umocnenia, $A(4,n)$ mocninovej veže, atď.

Vypočítateľnost

Intuitívne, priamočiary postup vyhodnocovania je konečný, lebo na výpočet A(niečo, niečo) potrebujeme vypočítať A(to isté, menej) a následne A(menej, whatever).

Pre nás je už predstava rekurzívneho programu prirodzená, za „starých čias“ museli nájsť nejaký postup bez rekurzie ako ju vyhodnocovať a museli o ňom dokázať, že je konečný. Zafunguje napr. postup: „Udržuj si zoznam hodnôt, ktoré si už vypočítal, a zoznam hodnôt, ktoré vypočítať chceš. V každom kole prejdi zoznam hodnôt, ktoré chceš vypočítať. Každú, ktorú už vieš, z neho vyhoď, a ku každej, ktorú nevieš, pridaj do zoznamu jej prerekvizitu.“

Nie primitívna rekurzívnosť

Platí: nech $f : \mathbb{N}^k \to \mathbb{N}$ je primitívne rekurzívna funkcia. Potom existuje konštanta Q_f taká, že

$$\forall a_1, \dots, a_k : \; f(a_1, \dots, a_k) < A(Q_f, \sum a_i)$$

Špeciálne teda aj pre unárne funkcie: ku každej unárnej primitívne rekurzívnej funkcii f existuje konštanta Q_f taká, že $\forall x : \; f(x) < \; A(Q_f, x)$.

Pre dôkaz to nie je podstatné, ale konštantu Q_f vieme k danej primitívne rekurzívnej funkcii konštruktívne určiť podľa jej „receptu“. Na formálny dôkaz bude linka na stránke.

Čo nám toto tvrdenie hovorí? Že k ľubovoľnej primitívne rekurzívnej funkcii f existuje v Ackermannovej funkcii „riadok“, ktorý je od f ostro väčší.

Predpokladajme sporom, že $A(m, n)$ je primitívne rekurzívna. Potom je primitívne rekurzívna aj $B(x) = A(x, x) + 1$. A keďže B je primitívne rekurzívna, existuje také Q_B, že $\forall x : \; B(x) < A(Q_B, x)$.

A keď pre všetky x, tak aj pre $x = Q_B$. Musí teda platiť $B(Q_B) < \; A(Q_B, Q_B)$. Lenže z definície B je $B(Q_B) = A(Q_B, Q_B) + 1$, a teda daná nerovnosť neplatí. A to je hľadaný spor.

(Myšlienka dôkazu: $A(m, n)$ rastie rýchlejšie ako hociktorá primitívne rekurzívna funkcia. Keby sama bola primitívne rekurzívna, musela by rásť rýchlejšie sama od seba, a to sa nedá.)

1.2 Kódovanie primitívne rekurzívnych funkcií

Každú primitívne rekurzívnu funkciu vieme zapísať ako reťazec nad konečnou abecedou, napr. sčítanie ako `PR[P(1,1),Comp[S,P(3,1)]]`. A takýto reťazec ľahko zakódujeme do čísla (napr. jednotlivé znaky interpretujeme ako cifry vo vhodnej báze).

Napr. keby sme zobrali ASCII kódovanie a interpretovali zápisy ako čísla v 256-kovej sústave, vyššie uvedený zápis sčítania by zodpovedal číslu

$$504\,187\,519\,039\,233\,506\,125\,110\,627\,154\,402\,584\,544\,263\,683\,536\,271\,492\,865\,373$$

Alternatívne, ku každej primitívne rekurzívnej funkcii existuje aspoň jeden recept, a každému receptu vieme priradiť postupnosť čísel – napr. tak, že každý krok popíšeme postupnosťou čísel a tieto potom zreťazíme.

Ako popísať jednotlivé možné kroky? Successora zapíšeme ako (1), zero ako (2), projekciu $P_(n, k)$ ako $(3, n, k)$, kompozíciu funkcií s číslami a a $b_1, \dots, b_n$ ako $(4, n + 1, a, b_1, \dots, b_n)$ a primitívnu rekurziu funkcií s číslami a a b ako $(5, a, b)$.

Pre sčítanie vieme spraviť napr. nasledujúci recept:

$$\begin{aligned} f_0 &= P_1^1 \\ f_1 &= s \\ f_2 &= P_1^3 \\ f_3 &= Comp[f_1, f_2] \\ f_4 &= PR[0, 3] \end{aligned}$$

Tomuto receptu by sme priradili postupnosť $(3, 1, 1, 1, 3, 3, 1, 4, 2, 1, 2, 5, 0, 3)$.

No a takúto postupnosť vieme zakódovať do prirodzeného čísla. Ak použijeme kódovanie do mocnín prvočísel, dostávame hodnotu:

$$20\,789\,288\,505\,826\,750\,568\,721\,059\,950\,644\,171\,480$$

Ak by sme namiesto toho použili opakovane párovaciu funkciu $s(c(x, y))$ (viď cvičenie 4.6), dostaneme hodnotu:

$$279\,228\,304\,593\,168\,697\,084 \ldots \text{(6060 cifier chýba)} \ldots\ 715\,629\,865\,406\,059\,540$$

Práve sme popísali tri rôzne spôsoby ako ľubovoľnej primitívne rekurzívnej funkcii priradiť prirodzené číslo, z ktorého ju vieme späť zostrojiť. Pre platnosť výsledkov v nasledujúcej časti je úplne jedno, ktoré z týchto kódovaní do čísel použijeme.

1.3 Univerzálna primitívne rekurzívna funkcia

Ku každému prirodzenému číslu n priraďme primitívne rekurzívnu funkciu f_n nasledovne: ak je n platným kódom (v nejakom pevnom, nami zvolenom kódovaní) nejakej unárnej primitívne rekurzívnej funkcie, tak f_n je táto funkcia, inak je f_n identicky rovná nule.

Funkcia $\Phi(n, x) = f_n(x)$ je algoritmicky vypočítateľná. (Viď program na webe.) Stačí si z čísla n zostrojiť predpis funkcie a ten postupne rekurzívne vyhodnocovať.

Funkcia Φ však nemôže byť primitívne rekurzívna – lebo $g(n) = \Phi(n, n) + 1$ by bola tiež. Lenže keď je g primitívne rekurzívna, má nejaké číslo y, teda $g \equiv f_y$. Potom ale $g(y) = \Phi(y, y) + 1 = f_y(y) + 1 = g(y) + 1$.

Funkciu $\Phi(n, x)$ voláme univerzálna funkcia pre unárne primitívne rekurzívne funkcie.